ORACLE®
FUSION APPLICATIONS
DEVELOPER RELATIONS

Oracle Sales Cloud Business Objects Cheatsheet

A quick reference guide for Groovy developers

ORACLE®

Table of Contents

Introduction

When you are developing customizations in Application Composer it is often required to write a short expression to add a condition that controls a specific feature, such as a trigger that updates another field only when a specific value is found, or to set when to fire an Object Workflow.

This is done using the Groovy scripting language and written in the Expression Palette region. This user interface offers tabs that help you in your coding, displaying common functions, keywords, pre-defined web services, and available fields. In practice many of the view objects that underpin the business objects contain complex and numerous fields and relationships, making writing accurate code sometimes a challenge. As in the screenshot below, without some prior knowledge or trial-and-error testing it would hard to know which of the Status or Win Probability fields to use.

The Fusion Applications Developer Relations team has put together this small document with a selection of ERD-like diagrams to help you understand the commonly used standard objects, their fields, and the relationships between them.

This document is intended as your coding companion, and at the very least it reduces the need to have multiple windows open to see the object information shown on different screens and pages. The content is not intended to be exhaustive, as to keep things simple we have included just the core object details.

The diagrams provide the primary/foreign key fields that control the object relationships, along with selection of API Names for the more useful fields. Where the purpose of the field is not obvious from the given API Name, a clue is provided in brackets. The information in this document is based on the Fusion Applications Release 8 version of the business objects, and whilst the structures may change in future releases it is likely most of the information will remain useful for some time. If you spot any mistakes or have suggestions for enhancements please contact us. If you are new to the Groovy language take a look at our Getting Started whitepaper.

The Opportunity Business Object

Found in the Sales application, this object represents converted leads or stand-alone sales process records. Being at the heart of the sales functionality, it has several child objects as well as seeded relationships to other standard objects.

Opportunity: Overview

⊿ **Object Information**

Display Label Opportunity
Plural Label Opportunities
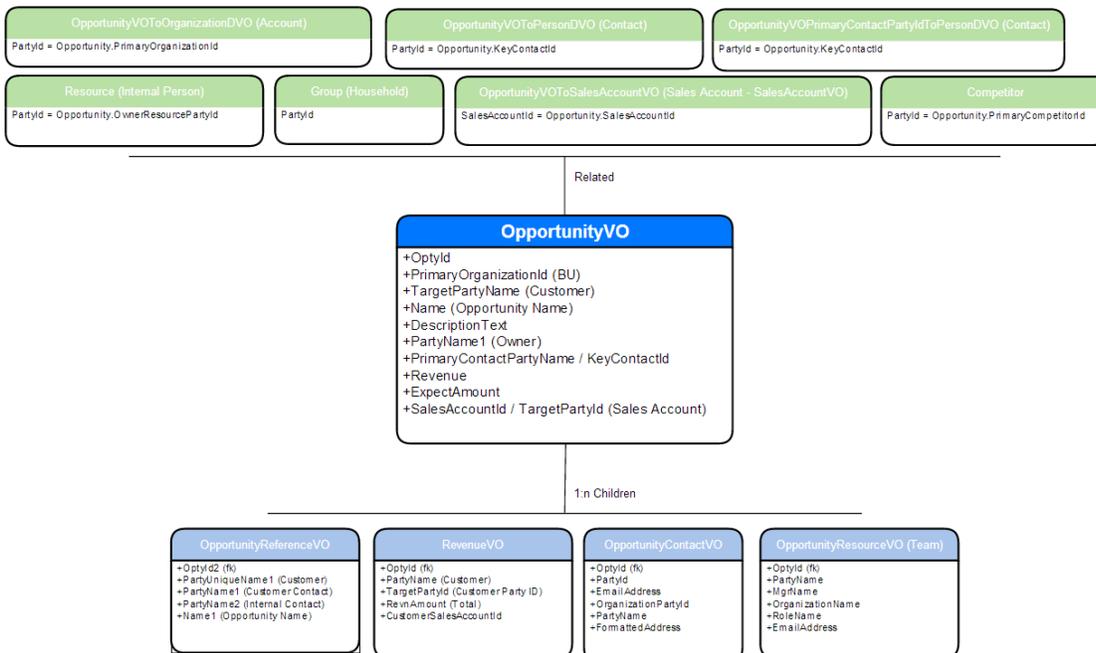Name OpportunityVO
API Name OpportunityVO

Child Objects

Opportunity Contact; Opportunity Reference; Opportunity Revenue; Opportunity Team Member

Related Objects

Account; Contact; Contact; Household; Relationship; Sales Account

The following diagram illustrates the Opportunity object fields and its supporting relationships.

OpportunityVOToOrganizationDVO (Account)	OpportunityVOToPersonDVO (Contact)	OpportunityVOPrimaryContactPartyIdToPersonDVO (Contact)
PartyId = Opportunity.PrimaryOrganizationId	PartyId = Opportunity.KeyContactId	PartyId = Opportunity.KeyContactId

Resource (Internal Person)	Group (Household)	OpportunityVOToSalesAccountVO (Sales Account - SalesAccountVO)	Competitor
PartyId = Opportunity.OwnerResourcePartyId	PartyId	SalesAccountId = Opportunity.SalesAccountId	PartyId = Opportunity.PrimaryCompetitorId

Related

OpportunityVO

+OptyId
+PrimaryOrganizationId (BU)
+TargetPartyName (Customer)
+Name (Opportunity Name)
+DescriptionText
+PartyName1 (Owner)
+PrimaryContactPartyName / KeyContactId
+Revenue
+ExpectAmount
+SalesAccountId / TargetPartyId (Sales Account)

1:n Children

OpportunityReferenceVO	RevenueVO	OpportunityContactVO	OpportunityResourceVO (Team)
+OptyId2 (fk)	+OptyId (fk)	+OptyId (fk)	+OptyId (fk)
+PartyUniqueName1 (Customer)	+PartyName (Customer)	+PartyId	+PartyName
+PartyName1 (Customer Contact)	+TargetPartyId (Customer Party ID)	+EmailAddress	+MgrName
+PartyName2 (Internal Contact)	+RevnAmount (Total)	+OrganizationPartyId	+OrganizationName
+Name1 (Opportunity Name)	+CustomerSalesAccountId	+PartyName	+RoleName
		+FormattedAddress	+EmailAddress

The Sales Account Business Object

Found in the Customer Center application, the Sales Account represents a unique implementation of the standard TCA Account object.

The following diagram illustrates the Customer Account object fields and its supporting relationships.
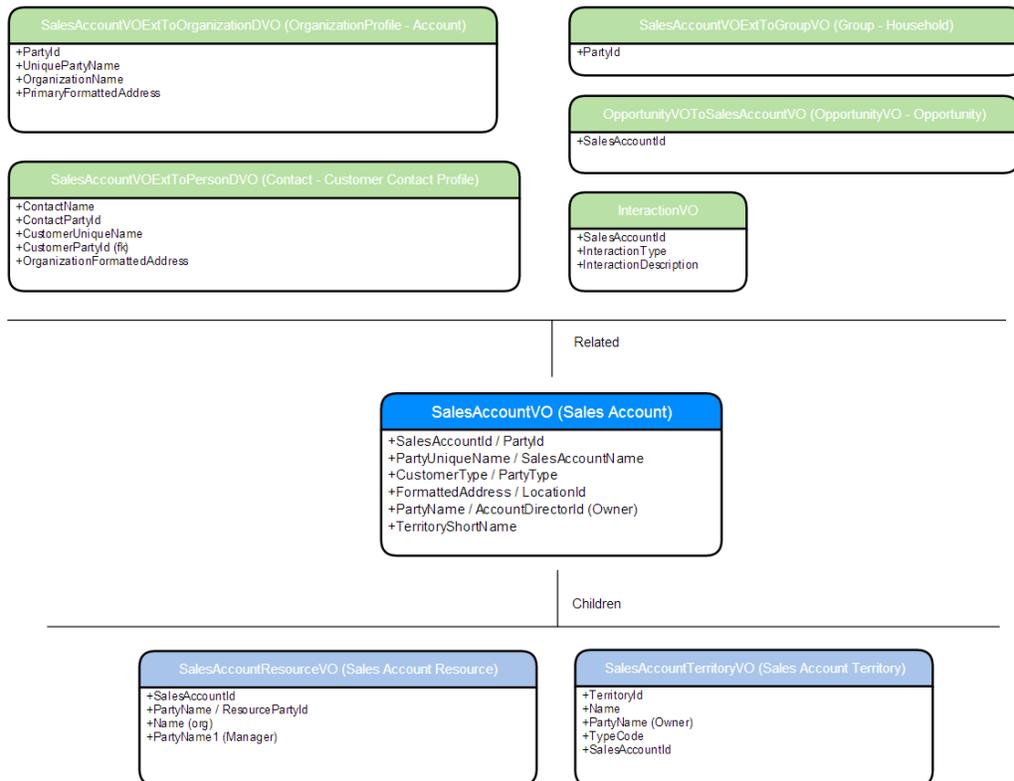
The Sales Lead Business Object

 Found in the Marketing application, the Sales Lead object is used across the sales features for capturing initial process information.

Sales Lead: Overview

Object Information

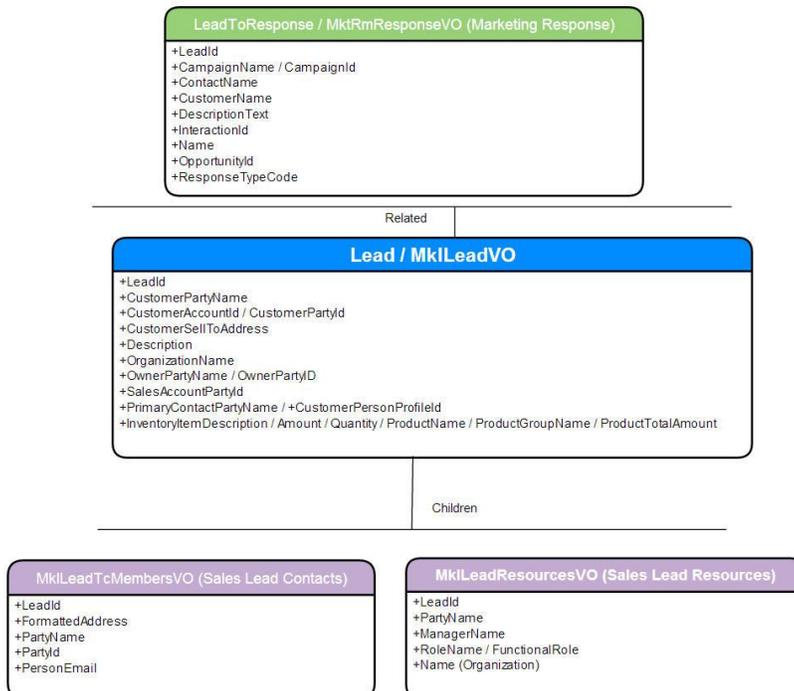|	
Display Label	Sales Lead
Plural Label	Sales Lead
Name	MklLeadVO
API Name	MklLeadVO

Child Objects

Sales Lead Contacts; Sales Lead Resources

Related Objects

Sales Lead does not have any related objects.

Whilst below shows no related objects, included in the ERD-like diagram is the internal view link to the Marketing Response object.

LeadToResponse / MktRmResponseVO (Marketing Response)
+LeadId
+CampaignName / CampaignId
+ContactName
+CustomerName
+DescriptionText
+InteractionId
+Name
+OpportunityId
+ResponseTypeCode

Related

Lead / MklLeadVO
+LeadId
+CustomerPartyName
+CustomerAccountId / CustomerPartyId
+CustomerSellToAddress
+Description
+OrganizationName
+OwnerPartyName / OwnerPartyID
+SalesAccountPartyId
+PrimaryContactPartyName / +CustomerPersonProfileId
+InventoryItemDescription / Amount / Quantity / ProductName / ProductGroupName / ProductTotalAmount

Children

MklLeadTcMembersVO (Sales Lead Contacts)
+LeadId
+FormattedAddress
+PartyName
+PartyId
+PersonEmail

MklLeadResourcesVO (Sales Lead Resources)
+LeadId
+PartyName
+ManagerName
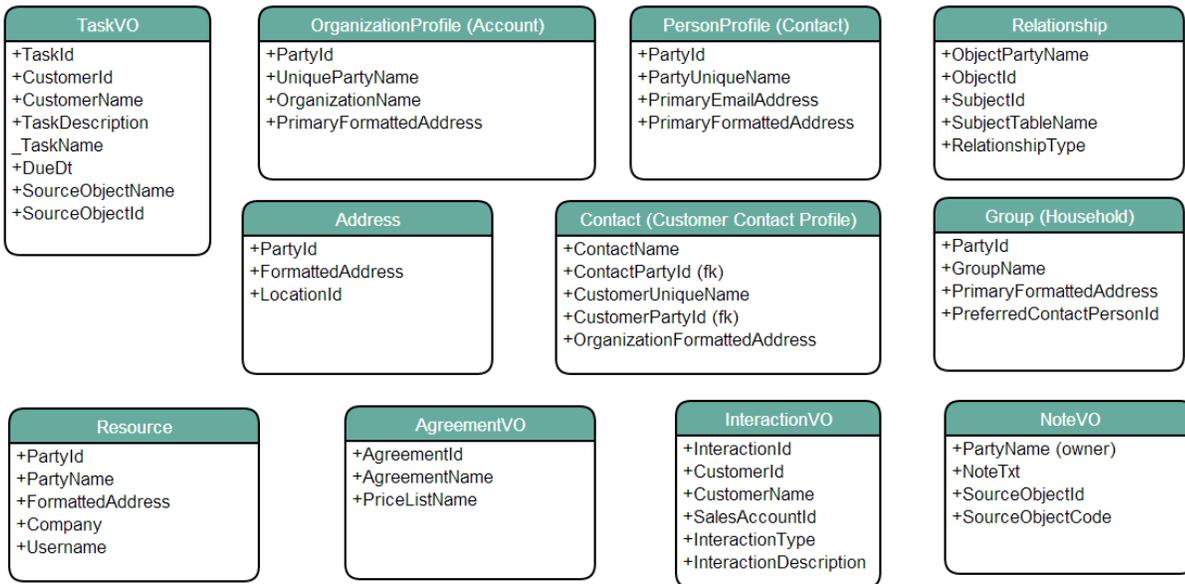+RoleName / FunctionalRole
+Name (Organization)

The Common Business Objects

In addition to product-specific objects, Application Composer exposes a set of objects which can be used to access more information for a particular record. These objects are found in the "Common" application and are more generic in their content, being derived from the underlying Fusion Applications data models, such as the Trading Community Architecture as shown here.

This means that whilst they are the root data entities they lose some of the CRM-specific context. An example might be the Account object which can have many different types, such as a customer (also implemented as the Customer Account object), a competitor, or a partner.

Read more about working with these objects in Chapter 11 of the Extending Sales guide.

TaskVO
+TaskId
+CustomerId
+CustomerName
+TaskDescription
_TaskName
+DueDt
+SourceObjectName
+SourceObjectId

OrganizationProfile (Account)
+PartyId
+UniquePartyName
+OrganizationName
+PrimaryFormattedAddress

PersonProfile (Contact)
+PartyId
+PartyUniqueName
+PrimaryEmailAddress
+PrimaryFormattedAddress

Relationship
+ObjectPartyName
+ObjectId
+SubjectId
+SubjectTableName
+RelationshipType

Address
+PartyId
+FormattedAddress
+LocationId

Contact (Customer Contact Profile)
+ContactName
+ContactPartyId (fk)
+CustomerUniqueName
+CustomerPartyId (fk)
+OrganizationFormattedAddress

Group (Household)
+PartyId
+GroupName
+PrimaryFormattedAddress
+PreferredContactPersonId

Resource
+PartyId
+PartyName
+FormattedAddress
+Company
+Username

AgreementVO
+AgreementId
+AgreementName
+PriceListName

InteractionVO
+InteractionId
+CustomerId
+CustomerName
+SalesAccountId
+InteractionType
+InteractionDescription

NoteVO
+PartyName (owner)
+NoteTxt
+SourceObjectId
+SourceObjectCode